An Algorithm for the Multiplication of Symmetric Polynomials

JOHN S. GARAVELLI
NASA Ames Research Center

Although the cycle index polynomial for a permutation group can often be easily determined, expansion of the figure counting series in a Pólya enumeration presents computational difficulties for object sets with higher degrees of symmetry and more than modest size. An algorithm that does not require algebraic symbol manipulation is derived for multiplying symmetric polynomials represented by partitions. Because the repetitive identification and collection of common terms are eliminated and storage requirements reduced, this algorithm is useful in rapidly expanding the figure counting series in such Pólya enumeration problems as the counting of chemical isomers.

Categories and Subject Descriptors: G.2.1 [**Discrete Mathematics**]: Combinatorics—*combinatorial algorithms*; I.1.1 [**Algebraic Manipulation**]: Expressions and Their Representation—*representations (general and polynomial)*; I.1.2 [**Algebraic Manipulation**]: Algorithms—*algebraic algorithms*

General Terms: Algorithms, Theory

Additional Key Words and Phrases: Cycle index polynomial, integer partitions, Pólya enumeration, symbolic algebra, symmetric polynomials

1. INTRODUCTION

For any permutation group, it is usually possible to determine quickly the appropriate cycle index polynomial. However, when the object set of the group has higher degrees of symmetry and more than modest size, substitution of the figure counting series in the cycle index polynomial and expansion in a Pólya enumeration [6] can present computational difficulties [7]. Manual solution is unfeasible, and computer programs relying on algebraic symbol manipulation are neither efficient nor readily accessible.

In most Pólya enumeration applications, the figure weights are merely formal variables, and the desired information is conveyed entirely by the coefficients and exponents of the formal variables. After substitution and expansion of the figure counting series in the cycle index polynomial, it is convenient to collect terms that occur into symmetric polynomials. A symmetric polynomial, which is the sum of all the terms with the same polynomial form (i.e., terms with the same set of exponents), can be compactly represented by a partition of the sum

Author's present address: Biomolecular Analysis Facility, College of Pharmacy (M/C 781), University of Illinois at Chicago, Box 6998, Chicago, IL 60680.

of the exponents, thus eliminating the formal variables. In order to achieve the economy this representation affords, an algorithm is required that provides the symmetric polynomials and their coefficients resulting from the multiplication of two given symmetric polynomials when the symmetric polynomials are represented by partitions. Such an algorithm could expand the cycle index polynomial in Pólya enumerations without repetitively identifying and collecting redundant terms, thus saving both computation time and storage. Additional savings in storage could be realized because there are efficient algorithms for numbering the partitions of an integer and for finding the partition of an integer corresponding to a given number [8]. By using these algorithms, it would not be necessary to store the partition representations of any intermediate results, but merely index numbers corresponding to them.

The formula that will be developed here for the multiplication of symmetric polynomials is similar to one reported by Lauer [3] with an unpublished proof attributed to R. Loos. However, the algorithms developed in that report did not take advantage of the partition representation of symmetric polynomials. In one case those algorithms employed a double sort procedure, which slowed the computations considerably, and in another case they failed to produce the correct, complete result for an example provided in that report.

2. TERMINOLOGY

A symmetric polynomial in ν variables will be represented by the set of the exponents of the variables.

$$(\zeta_1, \zeta_2, \ldots, \zeta_\mu; \nu) = \sum_{i_1=1}^{\nu} \sum_{i_2 \neq i_1} \cdots \sum_{i_\mu \neq i_1, i_2, \ldots, i_{\mu-1}} X_{i_1}^{\zeta_1} X_{i_2}^{\zeta_2} \cdots X_{i_\mu}^{\zeta_\mu},$$

such that no term is repeated. The elements in this set are most conveniently arranged in decreasing order, so that $\zeta_1 \geq \zeta_2 \geq \cdots \geq \zeta_\mu$. The sum of the elements in this set will be referred to as the order, δ, of the polynomial, and the ordered set can be regarded as a partition of δ. The number of elements in this set will be referred to as the degree of the polynomial, μ, so that $\mu \leq \nu$. If $\phi(z)$ is the number of ζ_i's equal to z with $z \in [1, \delta]$, then $\mu = \sum_{z=1}^{\delta} \phi(z)$, and $\delta = \sum_{z=1}^{\delta} z \cdot \phi(z)$. Let $[\phi^{\mu}(z)]$ represent $\mu!/\prod_{z=1}^{\delta} \phi(z)!$, the multinomial of μ over the set of $\phi(z)$ with $z \in [1, \delta]$. The number of nonidentical terms in the symmetric polynomial, which will be referred to as the multiplicity, is

$$\xi = \binom{\nu}{\mu}\begin{bmatrix}\mu \\ \phi(z)\end{bmatrix} = \frac{\nu!}{(\nu-\mu)! \cdot \prod_{z=1}^{\delta} \phi(z)!}.$$

When two symmetric polynomials are multiplied, the result can be represented as follows:

$$(\{\zeta_i\}_1; \nu_1) \times (\{\zeta_i\}_2; \nu_2) = \sum_{r=3}^{\rho+2} \gamma_r \cdot (\{\zeta_i\}_r; \nu_r).$$

The coefficients γ_r of the symmetric polynomials in the result will be referred to as the degeneracies. The number of unique symmetric polynomials in the result is ρ. The product of the multiplicities of the multipliers is equal to the sum of the products of the degeneracies and the multiplicities of the resulting symmetric polynomials; that is,

$$\xi_1 \cdot \xi_2 = \sum_{r=3}^{\rho+2} \gamma_r \cdot \xi_r.$$

In order to determine the general form of the result, a minimum of $\mu_1 + \mu_2$ variables must be used, so that $\nu_i \geq \mu_1 + \mu_2$ for $i \in [1, \rho + 2]$. In the following, it will be assumed that the required minimum value of ν is being employed and that its specification will be omitted from the partition representation of symmetric polynomials.

The partition sets representing the resulting symmetric polynomial exponents can be produced by selecting k exponents from the partition of each multiplier in each possible way, adding them together in each possible way, and appending the remaining exponents from the partition of both multipliers for each k from 0 to $\min(\mu_1, \mu_2)$. If $\rho_k = k!\binom{\mu_1}{k}\binom{\mu_2}{k}$, then the maximum possible value of ρ is given by $\sum_{k=0}^{\min(\mu_1,\mu_2)} \rho_k$.

In this notation, all multiplications with the form $(a + b) \times (a + b) = a^2 + b^2 + 2ab$ are represented by $(1) \times (1) = (2) + 2(1, 1)$, and all multiplications with the form

$$\begin{aligned}(a + b + c) &\times (a^2b + a^2c + b^2a + b^2c + c^2a + c^2b) \\ &= (a^3b + a^3c + b^3a + b^3c + c^3a + c^3b) \\ &\quad + 2(a^2b^2 + a^2c^2 + b^2c^2) + 2(a^2bc + b^2ac + c^2ab)\end{aligned}$$

are represented by $(1) \times (2, 1) = (3, 1) + 2(2, 2) + 2(2, 1, 1)$. It is for multiplications such as

$$\begin{aligned}&(5, 5, 5, 4, 3, 3) \times (1, 1) \\ &\quad = (6, 6, 5, 4, 3, 3) + 3(6, 5, 5, 5, 3, 3) + 2(6, 5, 5, 4, 4, 3) \\ &\quad\quad + (6, 5, 5, 4, 3, 3, 1) + 4(5, 5, 5, 5, 4, 3) + 4(5, 5, 5, 5, 3, 3, 1) \\ &\quad\quad + 3(5, 5, 5, 4, 4, 4) + 2(5, 5, 5, 4, 4, 3, 1) + (5, 5, 5, 4, 3, 3, 1, 1)\end{aligned}$$

that a rapid and accurate algorithm is needed [3].

3. SYMBOLIC MULTIPLICATION ALGORITHM

An algorithm for the multiplication of symmetric polynomials represented by partitions must provide the symmetric polynomials, represented by partitions, and the corresponding coefficients, referred to as the degeneracies, for each polynomial in the result. Begin with a straightforward procedure for the multiplication of symmetric polynomials using symbolic algebra:

Step 1. Calculate the multiplicity of both multipliers.
Step 2. Choose any term from the multiplier with the greater multiplicity, and multiply it by each term in the other multiplier.
Step 3. Count the terms that have the same ordered set of exponents, or form, in the result.
Step 4. For each distinct form in the result, divide its count by the multiplicity of the form, and multiply by the multiplicity of the first multiplier.

Consider the multiplication (1, 1) × (2, 1, 1). The minimum number of variables that must be used is 2 + 3 = 5, and the multiplicities would be 5!/3! 2! = 10 for (1, 1) and 5!/2! 1! 2! = 30 for (2, 1, 1). Choose the term a^2bc from (2, 1, 1), and multiply it by each term in the second multiplier, (1, 1):

$$a^2bc \times (ab + ac + ad + ae + bc + bd + be + cd + ce + de).$$

Count all the terms with the same ordered set of exponents, or form, in the result:

Form	Resulting terms	Term count	×	ξ_2	/	ξ_r	=	γ_r
(3, 2, 1)	$a^3b^2c + a^3c^2b$	2	×	30	/	60	=	1
(3, 1, 1, 1)	$a^3bcd + a^3bce$	2	×	30	/	20	=	3
(2, 2, 2)	$a^2b^2c^2$	1	×	30	/	10	=	3
(2, 2, 1, 1)	$a^2b^2cd + a^2b^2ce + a^2c^2bd + a^2c^2be$	4	×	30	/	30	=	4
(2, 1, 1, 1, 1)	a^2bcde	1	×	30	/	5	=	6

Therefore, (2, 1, 1) × (1, 1) = (3, 2, 1) + 3(3, 1, 1, 1) + 3(2, 2, 2) + 4(2, 2, 1, 1) + 6(2, 1, 1, 1, 1). As a check, the sum of the term counts must equal the multiplicity of the second multiplier.

The only step requiring algebraic symbol manipulation in this algorithm is Step 3, determination of the term counts by identifying the resulting terms that have the same form. In order to eliminate symbol manipulation, a formula is required for calculating the degeneracies that does not require determining these term counts.

THEOREM. *The coefficient of each symmetric polynomial in the result of the multiplication of two symmetric polynomials is given by*

$$\gamma_r = C_r \cdot \prod_{z=1}^{\delta} \frac{\phi_r(z)!}{\phi_1(z)!\ \phi_2(z)!},$$

where C_r is the number of ways the same ordered set of exponents can be produced when the exponents of the multipliers are combined in all possible ways.

Given two symmetric polynomials being multiplied, the degeneracies, γ_r, are directly calculated by choosing one term from the multiplier with the greater multiplicity, multiplying it by each term in the other multiplier, counting the terms that have the same form in the result, and dividing that count by the multiplicity of the form and multiplying by the multiplicity of the first multiplier. In order to count all the terms with the same form in the result, determine the k-term count, the number of terms in the second multiplier that have k variables in common with the chosen term of the first multiplier. Multiply that number by the form count, the number of ways the same set of ordered exponents can be produced when the exponents of the multipliers are combined in all possible ways, and divide by ρ_k, the number of ways the exponents of the multipliers can be combined when k of them are added.

To illustrate using the last example, there are three terms in the second multiplier that have two variables in common with the chosen term of the first multiplier, six terms that have one variable in common, and one term that has no variables in common:

$$a^2bc \times (\underbrace{ab + ac + bc}_{k=2} + \underbrace{ad + ae + bd + be + cd + ce}_{k=1} + \underbrace{de}_{k=0}).$$

	Form	k-Term count	$\times$	Form count	/	ρ_k	=	Term count
$k = 2$								
	(3, 2, 1)	3	$\times$	4	/	6	=	2
	(2, 2, 2)	3	$\times$	2	/	6	=	1
$k = 1$								
	(3, 1, 1, 1)	6	$\times$	2	/	6	=	2
	(2, 2, 1, 1)	6	$\times$	4	/	6	=	4
$k = 0$								
	(2, 1, 1, 1, 1)	1	$\times$	1	/	1	=	1

The k-term count, the number of terms in the second multiplier that have k variables in common with the chosen term of the first multiplier, is given by

$$\begin{bmatrix} \mu_2 \\ \phi_2(z) \end{bmatrix} \binom{\mu_2}{k} \sum_{i=0}^{k} (-1)^{i-k} \binom{k}{i} \binom{\mu_1 + i}{i}.$$

Using an identity from Gould [2], this reduces to

$$\begin{bmatrix} \mu_2 \\ \phi_2(z) \end{bmatrix} \binom{\mu_2}{k} \binom{\mu_1}{k}.$$

Therefore, if C_r is the form count, the degeneracy is given by

$$\gamma_r = \frac{C_r \binom{\mu_1}{k} \binom{\mu_2}{k} \left[\begin{smallmatrix} \mu_2 \\ \phi_2(z) \end{smallmatrix} \right] \xi_1}{\rho_k \xi_r},$$

which, upon substituting for ρ_k, k, ξ_1, and ξ_r, reduces to

$$\gamma_r = C_r \cdot \prod_{z=1}^{\delta} \frac{\phi_r(z)!}{\phi_1(z)!\ \phi_2(z)!}.$$

An algorithm that forms all the sets of exponents that can be produced when the sets of exponents of the two multipliers are combined and 0 through $\min(\mu_1, \mu_2)$ from each set are added in all possible permutations will also provide the required form counts, and thus the degeneracies, using this formula.

4. COMBINATORIC ALGORITHM

Using this last formula and the combinatorial subroutines "NEXKSB" and "NEXPER" from Nijenhuis and Wilf [5] to find the next selection of elements from a set and to permute the order of those elements, and "parttonum" and "numtopart" from Wells [8] to find the index corresponding to a partition and the partition corresponding to a number, it is possible to write an algorithm that will take two partition representations of symmetric polynomials and give the results that would be obtained if those symmetric polynomials were multiplied symbolically.

Step 1. Determine the order of the result: $N = \delta_1 + \delta_2$.
Step 2. $P(N)$ = number of partitions of N.
Step 3. Zero the counting array: $C(i) = 0$ for $i = 1$ to $P(N)$.
Step 4. *Loop* 1: for $k = 0$ to $\min(\mu_1, \mu_2)$.
Step 5. Pick the first subset of k exponents from the set of μ_1 exponents in the first multiplier.
Step 6. Pick the first subset of k exponents from the set of μ_2 exponents in the second multiplier.
Step 7. Pick the first permutation of the k exponents in the second subset.
Step 8. Add the k exponents from each subset together in order, and append the remaining exponents in both multipliers.
Step 9. Sort the resulting set of exponents, $\{\zeta_r\}$, in decreasing order.
Step 10. Determine the partition index, i, of the resulting set: i = "parttonum"($\{\zeta_r\}$).
Step 11. Increment the count for the partition index: $C(i) = C(i) + 1$.
Step 12. Pick the next permutation of the k exponents in the second subset, and return to Step 8 otherwise.
Step 13. Pick the next subset of k exponents from the second multiplier, and return to Step 7 otherwise.
Step 14. Pick the next subset of k exponents from the first multiplier, and return to Step 6 otherwise.
Step 15. Continue Loop 1.
Step 16. *Loop* 2: for $i = 0$ to $\min(\mu_1, \mu_2)$.
Step 17. Calculate the degeneracy: If $C(i) > 0$ then $C(i) = C(i) \cdot \prod_{z=1}^{\delta} [\phi_r(z)!/(\phi_1(z)!\ \phi_2(z)!)]$.
Step 18. Continue Loop 2.

5. COMPUTATION TIMES AND RESULTS

An analysis of the theoretical order of computation time for this algorithm indicates that it should proceed in times given by

$$\sum_{k=0}^{\min(\mu_1,\mu_2)} \binom{\mu_1}{k}\left[T_1 + \binom{\mu_2}{k}[T_1 + k![(k+1)T_2 + T_3 + T_4 + T_5]]\right],$$

where the component computation times are T_1 for "NEXKSB," forming the next subset of k items from either μ_1 or μ_2 items; T_2 for integer addition; T_3 for a sort of $\mu_1 + \mu_2 - k$ items; T_4 for "parttonum," determination of the index of a partition; and T_5 "NEXPER," forming the next permutation of k items. In the worst case, $\mu_1 = \mu_2 = \mu$ and the computation time would be proportional to

$$[2^\mu + (\pi\mu)^{-1/2}2^{2\mu}T_1] + \sum_{k=0}^{\mu} \binom{\mu}{k}\binom{\mu}{k} k![(k+1)T_2 + T_3 + T_4 + T_5].$$

If the components in the innermost loop were not dependent on k, the computation time for the dominating inner sum would be proportional to

$$\mu!(2\pi e z)^{-1/2}e^z \qquad \text{where} \quad z = 2\mu^{1/2}$$

Table I. *Computation Time for Multiplication Algorithm*

Order of result	Time (CPU sec)	Number of multiplications	Average time per multiplication
2	0.040	1	0.040
3	0.010	2	0.005
4	0.050	7	0.007
5	0.080	11	0.007
6	0.220	26	0.008
7	0.380	40	0.010
8	1.050	83	0.013
9	1.970	120	0.016
10	6.090	223	0.027
11	13.130	320	0.041
12	47.080	566	0.083
13	112.950	784	0.144
14	451.400	1,310	0.345
15	1,175.780	1,802	0.652
16	5,065.670	2,922	1.734
17	14,027.621	3,938	3.562
18	64,063.473	6,180	10.366

asymptotically as μ tends to infinity [4]. However, that is not the case, and the order of computation time can be expected to be slightly worse. Unlike the multiplication algorithm of Lauer, a double sort of terms is not required in the inner loop, and the computation time and results are entirely independent of ν, the total number of variables in the symmetrical polynomials.

This algorithm has been implemented as a FORTRAN subroutine MULPOL [1] in a program POLYA, which completely enumerates the permutation isomers of chemical compounds. Because of the savings in storage afforded by this subroutine, it was possible to enumerate the permutation isomers of compounds with more than four different substituents and with up to 20 substitution sites. Among the compounds whose permutation isomers were enumerated was dodecahedrane. Determination of the permutation isomer counts for the 637 different substituent compositions of dodecahedrane is equivalent to the problem of counting all the possible colorings of the vertices of the dodecahedron.

Table I provides the average computation times on a VAX 11/785 in batch operation for this algorithm for symmetric polynomial multiplication implemented in the FORTRAN subroutine MULPOL. The average computation time for polynomial multiplication with results from orders 2 through 18 were determined by recording the CPU time in seconds for all possible polynomial multiplications with results of each order and by dividing by the number of multiplications. The peak working set size did not exceed 962 pages of 512 bytes. For these calculations the output of multiplication results was eliminated, and only one record was output for each order with the summary information.

ACKNOWLEDGMENT

The author wishes to thank Uri N. Peled for helpful discussions in estimating the order of computation time.

REFERENCES

1. GARAVELLI, J. S., AND LEONARD, J. E. Improvements in the computer enumeration of permutation isomers. *Comput. Chem. 9*, 2 (1985), 133–147.
2. GOULD, H. W. *Combinatorial Identities, Revised Edition.* Morgantown, W.V., 1972, p. 27, eq. 3.38.
3. LAUER, E. Algorithms for symmetrical polynomials. In *SYMSAC 76: Proceedings of the 1976 ACM Symposium on Symbolic and Algebraic Computation*, R. D. Jenks, Ed. ACM, New York, 1976, pp. 242–247.
4. LIFSCHITZ, V., AND PITTEL, B. The number of increasing subsequences of the random permutation. *J. Comb. Theory Series A 31*, 1 (July 1981), 1–20.
5. NIJENHUIS, A., AND WILF, H. S. *Combinatorial Algorithms.* Academic Press, New York, 1975, pp. 21–34, 49–59.
6. PÓLYA, G. Kombinatorische anzahlbestimmungen für gruppen, graphen und chemische verbindungen. *Acta Math. 68* (1937), 145–254. (Note: Since the submission of this work, a translation of this reference has appeared in PÓLYA, G., AND READ, R. C. *Combinatorial Enumeration of Groups, Graphs and Chemical Compounds*. Springer-Verlag, New York, 1987, pp. 1–95.)
7. READ, R. C. The enumeration of acyclic chemical compounds. In *Chemical Applications of Graph Theory*, A. T. Balaban, Ed. Academic Press, London, 1976, Chap. 4, pp. 25–61.
8. WELLS, M. B. *Elements of Combinatorial Computing.* Pergamon, Oxford, 1971, p. 152.

Received December 1985; revised July 1988; accepted July 1988